Utime

Vulnerable to TOCTOU issues

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-23

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6245 bytes

Attack Category	• Path spoofing or confusion problem • File Manipulation
Vulnerability Category	• Indeterminate File/Path • TOCTOU - Time of Check, Time of Use
Software Context	• File Management
Location	• unistd.h
Description	utime() and utimes() are functions which allow the last-accessed and last-modified timestamps of files to be changed. These will NOT change the last-changed timestamp. This program is at risk for abuse if it is a setuid or setgid program. If the file referenced by a call to one of these functions is specified by the user and steps are not taken to verify that the user has permission to alter the timestamps, a potential for abuse exists.
APIs	**Function Name** / **Comments** utime() / utimes() /
Method of Attack	The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results.
Exception Criteria	If proper checking is performed or user-specified input is not used, this is not a problem.
Solutions	**Solution Applicability** / **Solution Description** / **Solution Efficacy**

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

When the futimes() function or a similar function is available.	If you have the option of using a version of utimes() that operates on a file descriptor, such as futimes, you should use this as it will protect against TOCTOU problems.	Elimination of threat.
When user specification of the file to be altered is not necessary.	Do not rely on user-specified input to determine what file's timestamps will be altered.	This will reduce exposure but will not eliminate the problem.
When the file being altered is owned by the current user and group.	Set your effective gid and uid to that of the current user and group when executing this statement.	This will prevent an attacker from altering any file they can't already alter.
Generally applicable.	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.
Generally applicable.	Limit the interleaving of operations on files from	Does not eliminate the underlying vulnerability

		multiple processes.	but can help make it more difficult to exploit.
	Generally applicable.	Limit the spread of time (cycles) between the check and use of a resource.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Generally applicable.	Recheck the resource after the use call to verify that the action was taken appropriately.	Effective in some cases.

Signature Details	int utime(const char *filename, struct utimbuf *buf); int utimes(char *filename, struct timeval *tvp);
Examples of Incorrect Code	
Examples of Corrected Code	
Source References	• Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, ch 9 • ITS4 Source Code Vulnerability Scanning Tool[2] • utime() man page[3] • utimes() and futimes() man page[4]
Recommended Resource	
Discriminant Set	**Operating System**: • UNIX (All) **Languages**: • C • C++

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com